



SEI Cyber Minute Script & Visuals

Title: SCAIFE: An Alert Auditing Classification Prototype

Ebonie McNeil

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Document Markings

VIDEO/Podcasts/vlogs This video and all related information and materials ("materials") are owned by Carnegie Mellon University. These materials are provided on an "as-is" "as available" basis without any warranties and solely for your personal viewing and use. You agree that Carnegie Mellon is not liable with respect to any materials received by you as a result of viewing the video, or using referenced web sites, and/or for any consequence or the use by you of such materials. By viewing, downloading and/or using this video and related materials, you agree that you have read and agree to our terms of use (<http://www.sei.cmu.edu/legal/index.cfm>).

DM19-0681

Cyber Minute Script

Hello, this is Ebonie McNeil from CERT with your SEI Cyber Minute. Secure software is a top priority for many organizations. Static analysis tools output alerts that identify potential flaws in code. Last year, we publically published SCALE, an aggregator tool that displays various tool alerts and gives auditors an intuitive and user-friendly interface. **[Slide 4]** Auditing manual alert determinations can often be time consuming. The Source Code Analysis Integrated Framework Environment (commonly known as SCAIFE) is a multi-server software architecture and open source prototype system developed to integrate with SCALE and other similar tools to enable automatic alert classification and advanced prioritization. **[Slide 5]** The SCAIFE prototype is intended to be used by engineers and analysts who manually audit alerts. SCAIFE provides automatic alert classification using Machine Learning which gives a level of confidence that the alert is true or false. **[Slide 6]** The SCAIFE prototype also enables organizations to apply formulas that prioritize static analysis alerts by using factors they care about. **[Slide 7]** SCAIFE relies on REST-API principles to provide an interface for developers to quickly integrate and communicate with the server system. We have published a YAML-formatted file specifying the SCAIFE API, available at the CMU-SEI GitHub site for free downloads by the public. **[Slide 8]** The YAML specification provides the SCAIFE API definition beta version in a format that developers can easily use to view, modify, and automatically generate code from. Thanks for watching this SEI Cyber Minute. For more information, please visit our website or send me an email at info@sei.cmu.edu.

SCALE Interface

Header Menu

Project: dos2unix

Fused View: On

Alert Filters

New Diagnostic

All IDs: Verdict: Previous: Path:

Line: Checker: Tool: Condition: Taxonomy:

Sort direction: Sort by:

Middle Menu

Showing 1 to 10 of 253 | Diagnostics per page: Go

Set selected to: Classifier Selected: None Selected

← Previous 1 2 3 4 5 6 7 8 9 ... 25 26 Next →

Alert List

ID	Flag	Verdict	Supplemental	Notes	Previous	Path	Line	Message	Checker	Tool	Condition	Title	Confidence	Alert Pri	Sev	Lik	Rem	Pri	Lev	CWE_Lik
1012	[I]	[Unknown]	Edit	0	0	/src/common.c	732	Assignment of function parameter has no effect outside the function. Did you forget dereferencing it?	uselessAssignmentPtrArg	cppcheck	CWE-398	N/A	-	-						N/A
1013	[I]	[Unknown]	Edit	0	0	/src/common.c	772	Assignment of function parameter has no effect outside the function. Did you forget dereferencing it?	uselessAssignmentPtrArg	cppcheck	CWE-398	N/A	-	-						N/A
1009	[I]	[Unknown]	Edit	0	0	/src/common.c	799	Condition '!RetVal' is always true	knownConditionTrueFalse	cppcheck	CWE-570	N/A	-	-						N/A
1010	[I]	[Unknown]	Edit	0	0	/src/common.c	799	Condition '!RetVal' is always true	knownConditionTrueFalse	cppcheck	CWE-571	N/A	-	-						N/A
1011	[I]	[Unknown]	Edit	0	0	/src/common.c	1838	Variable 'RetVal' is assigned a value that is never used.	unreadVariable	cppcheck	CWE-563	N/A	-	-						N/A
1003	[I]	[Unknown]	Edit	0	0	/src/common.c	141	The scope of the variable 'errstr' can be reduced.	variableScope	cppcheck	DCL19-C	Minimize the scope of variables and functions	-	-	1	1	2	2	3	
1005	[I]	[Unknown]	Edit	0	0	/src/common.c	199	The scope of the variable 'errstr' can be reduced.	variableScope	cppcheck	DCL19-C	Minimize the scope of variables and functions	-	-	1	1	2	2	3	
1006	[I]	[Unknown]	Edit	0	0	/src/common.c	544	The scope of the variable 'bomr' can be reduced.	variableScope	cppcheck	DCL19-C	Minimize the scope of variables and functions	-	-	1	1	2	2	3	
1001	[I]	[Unknown]	Edit	0	0	/src/common.c	732	Assignment of function parameter has no effect outside the function. Did you forget dereferencing it?	uselessAssignmentPtrArg	cppcheck	MSC12-C	Detect and remove code that has no effect	-	-	1	1	2	2	3	
1002	[I]	[Unknown]	Edit	0	0	/src/common.c	772	Assignment of function parameter has no effect outside the function. Did you forget dereferencing it?	uselessAssignmentPtrArg	cppcheck	MSC12-C	Detect and remove code that has no effect	-	-	1	1	2	2	3	

Source Code Viewer

src

Last updated Wed Aug 22 22:09:30 EDT 2018

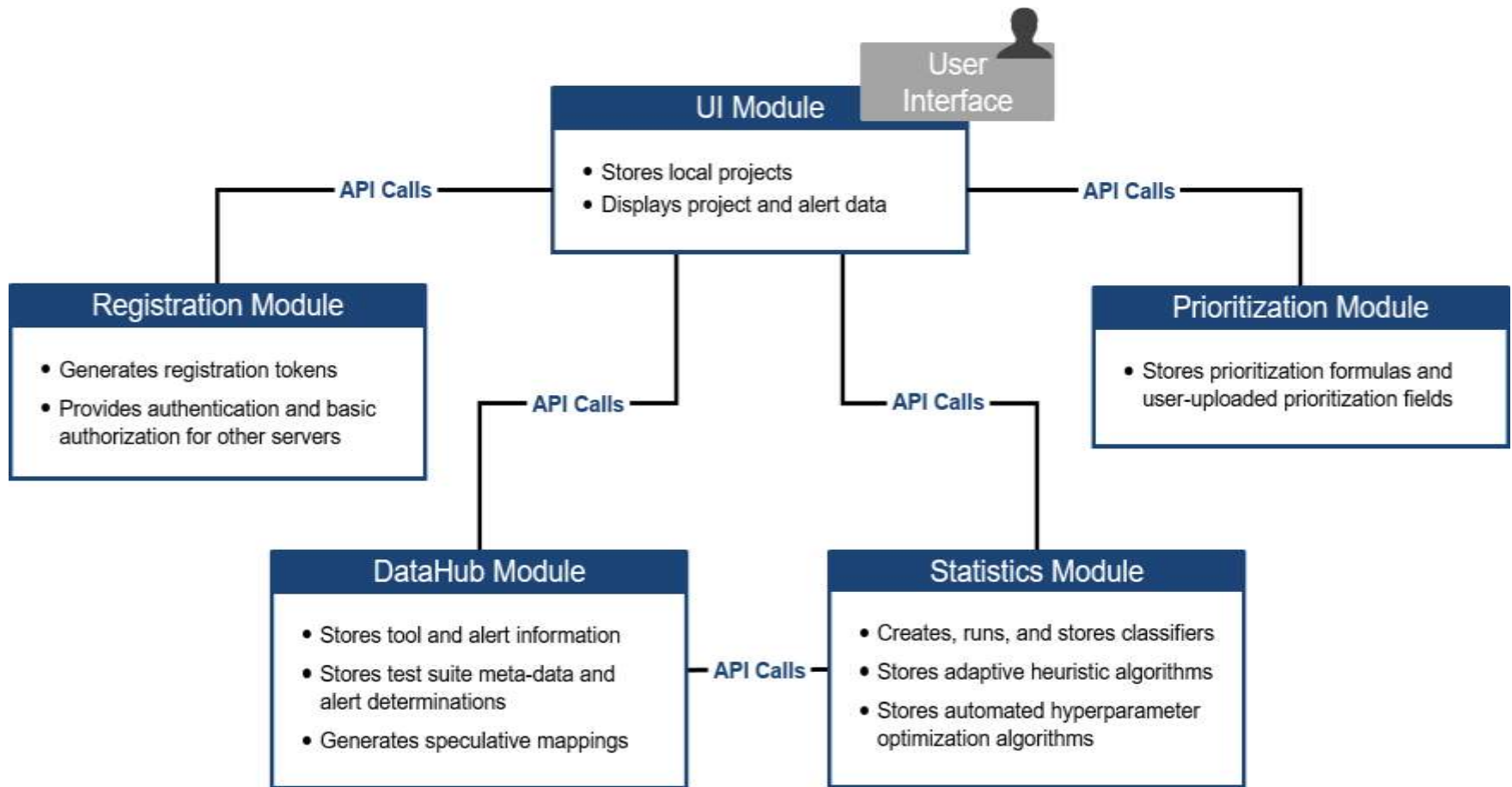
MAINS

```

#010      448 src/dos2unix.c int main (int argc, char *argv[])
#010      191 src/querycp.c int main() {
#010      8 src/test2cstombs_test.c int main() {

```

SCAIFE Architecture



Automatic Alert Classification

Select 'Classify' button to run the classifier on a project

- Classifier predicts alert determinations
- Meta-alerts will be classified
- Currently, example metrics are loaded for the 'Confidence' field
 - Usability demonstration only
 - Values not currently from classifier



	Confidence	Alert Pri	Sev	Lik
	4.85			
	30.28			
	91.08			
	84.84			
	1.68			
ns	91.91		1	1
ns	83.26		1	1

Alert Prioritization

Prioritization schemes with mathematical formulas user can create and/or use



Create New Scheme

Name:

Instructions CWES CERT_RULES

cert_severity

cert_likelihood

cert_remediation

cert_priority

cert_level

confidence

Formula for CERT_RULES

Generate The Formula

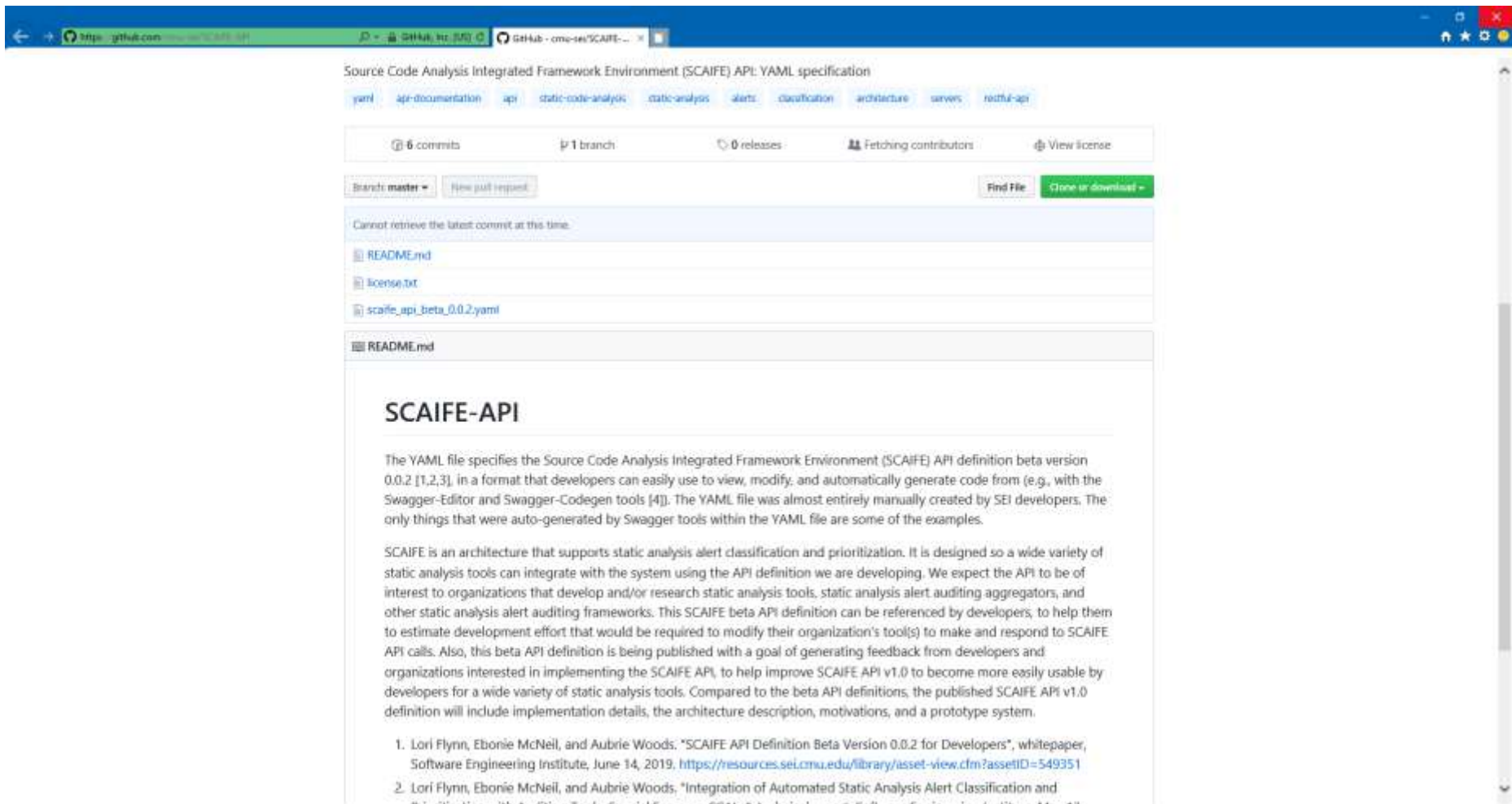
Prioritization Formula:

Save Priority Priority Scheme Saved

Cancel Run Priority

Confidence	Alert Pri	Sev	Lik
	83.48		
	33.12		
	15.27		

SCAIFE API on GitHub



The screenshot shows the GitHub repository page for the SCAIFE API: YAML specification. The repository is named "Source Code Analysis Integrated Framework Environment (SCAIFE) API: YAML specification" and is located in the "cmu-sei/SCAIFE" organization. The repository has 6 commits, 1 branch, and 0 releases. The current branch is "master". The repository contains files: README.md, license.txt, and scaife_api_beta_0.0.2.yaml. The README.md file is selected and its content is displayed below. The content of the README.md file is as follows:

SCAIFE-API

The YAML file specifies the Source Code Analysis Integrated Framework Environment (SCAIFE) API definition beta version 0.0.2 [1,2,3], in a format that developers can easily use to view, modify, and automatically generate code from (e.g., with the Swagger-Editor and Swagger-Codegen tools [4]). The YAML file was almost entirely manually created by SEI developers. The only things that were auto-generated by Swagger tools within the YAML file are some of the examples.

SCAIFE is an architecture that supports static analysis alert classification and prioritization. It is designed so a wide variety of static analysis tools can integrate with the system using the API definition we are developing. We expect the API to be of interest to organizations that develop and/or research static analysis tools, static analysis alert auditing aggregators, and other static analysis alert auditing frameworks. This SCAIFE beta API definition can be referenced by developers, to help them to estimate development effort that would be required to modify their organization's tool(s) to make and respond to SCAIFE API calls. Also, this beta API definition is being published with a goal of generating feedback from developers and organizations interested in implementing the SCAIFE API, to help improve SCAIFE API v1.0 to become more easily usable by developers for a wide variety of static analysis tools. Compared to the beta API definitions, the published SCAIFE API v1.0 definition will include implementation details, the architecture description, motivations, and a prototype system.

1. Lori Flynn, Ebonie McNeil, and Aubrie Woods, "SCAIFE API Definition Beta Version 0.0.2 for Developers", whitepaper, Software Engineering Institute, June 14, 2019, <https://resources.sei.cmu.edu/library/asset-view.cfm?assetID=549351>
2. Lori Flynn, Ebonie McNeil, and Aubrie Woods, "Integration of Automated Static Analysis Alert Classification and Prioritization into the Source Code Analysis Integrated Framework Environment (SCAIFE) Technical Report, Software Engineering Institute, June 14, 2019, <https://resources.sei.cmu.edu/library/asset-view.cfm?assetID=549352>